IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION

FOR

UNITED STATES PATENT

FOR

SYSTEM AND METHOD FOR GROUP VIDEO TELECONFERENCING WITH VARIABLE BANDWIDTH

INVENTORS:

Percy L. Spencer

Max E. Montgomery

Petrus H. Weyzen

Jeremy Egenberger

Don Fossgreen

BACKGROUND

Field of the Invention

The present invention relates to video-teleconferencing, and more particularly to dynamic allocation of bandwidth during video-teleconferencing.

Background of the Invention

Current video teleconferencing technology is plagued with comparatively high latency, low efficiency, and poor scalability. One reason for this is that current technologies use a

5   "lowest common bandwidth" method for determining the speed of transmission and packet size. Thus, if multiple clients are conferencing simultaneously, the transmission of the video data is only as fast as the lowest bandwidth will allow. As a result, in a conference in which some clients are using relatively slow dialup connections, while others are using T1, DSL, or similar broadband connections, those clients using broadband connections will receive data only at the

10   rate of the dialup connection, thus under utilizing their capabilities.

Current video teleconferencing techniques use the store and forward method for transmitting video frames. As video frames are generated, they are stored in their entirety on the generating computer. The frames are then forwarded to the server where they are again stored in their entirety and forwarded to the receiving computer. This requires large amounts of available

15   memory on the server and increases the workload of the server. As a result, conventional systems have poor scalability and increased latency.

Current video teleconferencing techniques often encounter difficulties when trying to pass through a firewall or proxy server. Firewalls are not compatible with data sent using UDP (User Datagram Protocol), a protocol that is commonly used by video teleconferencing

20   technologies. Proxy servers are used to filter requests and as a result, may filter out certain types of traffic often including video conferencing traffic.

In view of the foregoing limitations, there is a need for a video teleconferencing system that takes better advantage of the bandwidth capabilities of all clients, reduces required memory and load on the processing servers and is compatible with firewalls and proxy servers.

SUMMARY OF THE INVENTION

The present invention reduces latency and increases efficiency of multimedia group

5   conferencing by providing a system for dynamically transmitting data that includes a tiered-

server architecture. Clients using the system for multimedia group conferencing are connected to

a network and transmit and receive audio and video data via the network. When a client accesses

the system, one of the servers determines the maximum bandwidth available for the connection

to that client. The server then establishes an appropriate rate of transmission and packet size of

10   the data being transmitted in order to take full advantage of the available bandwidth. During the

transmission of the multimedia data, the server continues to monitor the bandwidth of the

connection to the client and adjusts the rate of transmission and packet size in response to

changes in the bandwidth.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a network diagram including an exemplary embodiment of the present invention.

5   Figure 2 is a multimedia streaming diagram in accordance with an exemplary embodiment of the present invention.

Figure 3 is a block diagram of a room server according to an exemplary embodiment of the present invention.

Figure 4 is a block diagram of a client according to an exemplary embodiment of the

10   present invention.

Figure 5 is a diagram of a threading model according to an exemplary embodiment of the present invention.

Figure 6 is a flow chart of dynamic data transmission according to an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Figure 1 is a diagram of an exemplary embodiment of a system including the present invention. The system includes a network 100, a router 112, one or more clients 102, and one or

5   more servers 104. In a preferred embodiment, two or more of the clients 102 send and receive multimedia data to each other via the network 100. The servers 104 facilitate any multimedia functionality that may be required for the accurate transmission of the data from client to client. The router 112 may be any commonly used routing device that facilitates the data flow to and from the servers 104. In a preferred embodiment, a tiered-server architecture includes some or

10   all of entry servers 106, lobby servers 108, and room servers 110 (collectively, servers 104.) The metaphor of lobbies and rooms facilitates load balancing and a place-oriented conferencing environment. Instead of choosing to conference with individuals, each client 102 may choose to enter a lobby and a room within that lobby. Similar to an online chat room, each client 102 is able to send audio, video and data to one or more other clients within a room.

15   The servers 104 are connected to the clients 102 via the network 100. In a typical embodiment, the network 100 may be the Internet, a proprietary network or an intranet, however other networks may also be used and the particular form of network is not limiting. Alternately, in some embodiments, the servers 104 and clients 102 may communicate indirectly or directly without passing through the network 100. The client 102 may have any number of

20   configurations of audio and video equipment to facilitate sending and receiving audio and video signals. This equipment may include a video display unit, speakers, a microphone, a camera, and a processing unit running suitable software to implement the conferencing functionality described below. An exemplary configuration of a client 102 is described in greater detail with the discussion of Fig. 4, below.

To send and receive multimedia data, clients 102 exchange information with servers 104. An exemplary embodiment includes one or two entry servers 106, however, the system is not limited to this number of entry servers 106. The entry servers 106 are responsible for the administrative functionality of logging-in clients 102, which includes providing password

5   encryption during the log-in process. The entry servers 106 are also responsible for maintaining a directory of available lobbies, allowing each client 102 to choose a lobby, and ensuring that that client 102 has permission to enter that lobby. The entry servers 106 are easily clustered, since the only state information contained in the entry servers 106 is the directory of available lobbies. The entry servers 106 also assist in the client-initiated analysis of bandwidth, latency,

10   and protocol availability. When a client logs in, the client 102 and entry server exchange a test transmission that together with other requested information establishes the bandwidth of the connection to and from the client 102 and determines whether UDP will work as a transmission protocol. If the use of UDP is not restricted by firewalls or proxy servers, then future transmissions during the session will be sent using UDP. If, however, the use of UDP is

15   restricted, then future transmissions will be sent using TCP (transmission control protocol.)

The lobby servers 108 send identifying information to the entry servers 106. This information includes a list of clients that do not have access to the lobby. The lobby servers 108 also perform a load balancing function. If a client 102 requests the creation of a new room, the lobby server 108 creates the room on the room server 110 that has the least load. In a preferred

20   embodiment, any client 102 that is logged into a lobby may request the creation of a new room. Alternatively, the creation of new rooms may be restricted to predetermined clients 102 or clients that fulfill certain criteria. For instance, requesting the creation of a new room may be restricted to those clients 102 who have provided billing information such that the use of the room by any

client 102 may be charged to the creating client 102. In an exemplary embodiment, the client

102 requesting the creation of a new room, or the moderator, is assigned special control

privileges over the conference. For example, the moderator may prevent certain clients 102 from

continuing to participate in the conference, may control which clients 102 have access to certain

5   types of information, or may close the room. Moderators may also delegate the special

privileges to another client 102. In a preferred embodiment, a lobby server 108 may support a

plurality of room servers 110, for example up to seven or more room servers 110. From the

lobby, a client 102 has an option of requesting the creation of a new room or entering an existing

room.

    In a preferred embodiment, the room servers 110 facilitate the multimedia functionality

10   of the system. The room server 110 is discussed in greater detail in the description of Figure 3,

below. Figure 1 shows only one example of a possible architecture and the invention is not

limited to the exemplary architecture illustrated in Figure 1.

    Figure 2 is a multimedia streaming diagram in accordance with an exemplary

embodiment of the present invention. The clients 102A, 102B, 102N (collectively clients 102)

15   exchange audio and video data with each other via the room server 110. Each client 102 may

include a transmitter 204 and a receiver 202. The room server 110 establishes a unique receiver

210 and transmitter 212 for each client 102 that is transmitting data through the room server 110.

The clients 102 are connected to the room server 110 via a network 100, not shown in Figure 2.

20   The clients 102 and room server 110 are described in greater detail in the discussion of Figures 3

and 4, below.

    The audio data 216 and video data 214 are sent from the transmitter 204 of the generating

client 102 to the receiver 210 for that client 102 at the room server 110. In a preferred

embodiment, each client 102 chooses which video and audio to view and hear. These choices are facilitated through the use of subscriber lists and subscription lists. The subscriber lists are used in conjunction with receivers 202, 210 to redistribute data to other clients in a room. Each receiver 202, 210 is grouped with one subscriber list for audio data and one subscriber list for

5   video data. The subscriber list identifies those clients who have subscribed to a given audio stream or video stream. The subscription list is used in conjunction with the transmitters 204, 212 to correlate video streams with specific video channels so that this data can be multiplexed. Each transmitter 204, 212 is grouped with one subscription list for audio and one subscription list for video. The subscription list identifies those clients whose audio and video will be transmitted

10   to a given client. In an exemplary embodiment, the audio subscription list may contain only one entry since each client 102 may hear only one audio stream at a time. The video subscription list may contain up to eight entries, one for each video window that may be simultaneously displayed.

Based on the information in the subscriber lists and subscription lists, the receivers 210 in

15   the room server 110 send video and audio streams 214, 216 to the transmitters 212 of the receiving clients 102 in the room server 110. The transmitters 212 then send the video and audio to the respective clients 102. The transmission of the multimedia data is discussed in greater detail in the description of Figures 3 and 4, below.

In the example shown in Figure 2, client 102A is transmitting video data 214A and audio

20   data 216A. The other two clients shown, clients 102B and 102N are transmitting video data 214B and 214N respectively. Client 102A is receiving its own video 214A and video 214B from client 102B. As a result, the video subscription list for transmitter 212A will contain clients 102A and 102B, and the video subscriber lists for both receiver 202A and 202B will contain

client 102A. Note that in the embodiment shown, the video 214A of client 102A is transmitted over the network 100 to the room server 110 and back. In an alternate embodiment, client 102A may view a local video image as direct feedback without video 214A being transmitted over the network and back. Client 102B is receiving video 214A and audio 216A from client 102A and

5 video 214N from client 102N. Client 102N is receiving video 214A and audio 216A from client 102A and video 214B from client 102B. When clients 102B and 102N request to see and hear this audio and video data, the relevant subscription and subscriber lists are updated.

Transmitter 204A at client 102A sends the audio stream 216A and video stream 214A generated at client 102A to receiver 210A at the room server 110. Receiver 210A sends the

10 audio stream 216A to transmitter 212B and transmitter 212N for transmission to clients 102B and 102N respectively. Receiver 210A sends the video stream 214A to transmitters 212A, 212B, and 212N for transmission to clients 102A, 102B, and 102N respectively. Transmitter 204B at client 102B sends the video stream 214B generated at client 102B to receiver 210B at the room server 110. Receiver 210B sends the video stream 214B to transmitters 212A and 212N for

15 transmission to clients 102A and 102N respectively. Transmitter 204N sends video stream 214N generated at client 102N to receiver 210N at the room server 110. Receiver 210N sends the video stream 214N to transmitter 212B for transmission to client 102B.

Transmitter 212A sends video 214A and 214B to receiver 202A at client 102A. Transmitter 212B sends video 214A and 214N and audio 216A to receiver 202B at client 102B.

20 Transmitter 212N sends video 214A and 214B and audio 216A to receiver 202N at client 102N. These transmissions from transmitters 212A, 212B, 212N are governed by the respective subscription lists for those transmitters.

In addition to video and audio transmissions, the clients may also transmit data such as slide show presentations, text documents, photographic images, music files, etc. Like the video and audio streams depicted in Figure 2, the data stream may be sent from any client 102 to one or more receiving clients 102. Figure 2 depicts three clients 102, however, there may be any

5   number of clients 102 each with a unique transmitter 212 and receiver 210 at the room server 110.

Figure 3 is a block diagram of a room server according to an exemplary embodiment of the present invention. The room server 110 may include zero, one or more pairs of receivers 210 and transmitters 212. In a preferred embodiment, the receiver 210 and transmitter 212 are

10   implemented in software and the room server 110 creates a unique receiver 210 and transmitter 212 for each client 102 that is sending or receiving multimedia data. The receiver 210 may include a sequencer 306. The transmitter 212 may include some or all of an audio resequencer 308, a video resequencer 310, a multimedia audio queue 312, a video multiplexer 314, and a packet encoder 316.

15   Each receiver 210 is connected to the network 100 to receive multimedia data from a client 102. The receiver 210 is also connected to one or more transmitters 212. The receiver 210 transfers the data received from the client 102 to the transmitter 212. The transmitter 212 is also connected to the network 100 and data transferred by the receiver 210 to the transmitter 212 is transmitted over the network 100 to the receiving client 102.

20   The room server 110 receives data in the form of multimedia blocks from the sending client 102. In a preferred embodiment, a multimedia block is a type of data packet that includes some or all of a sequence number, audio frames, video fragments, a video channel, a receipt, video parameters, audio parameters, a video end flag, and an audio end flag. The sequence

number is used to reorder the multimedia blocks if they contain audio or video data. If the

multimedia block contains audio data, this data would be in the form of an audio frame. If the

multimedia block contains video data, this data would be in the form of a video fragment. The

video fragment is a data structure that may represent the start, middle, or end of a video frame.

5   The video fragment may also be an entire video frame or a special value indicating that a video

fragment has been lost during a prior transmission. The video channel is the channel assigned to

the video fragment, if there is video data. The receipt is the sequence number of the most recent

multimedia block received by the other party. The receipt is used in determining the allocation

of bandwidth as discussed in the description of Figure 6, below. The video and audio parameters

10   are transmitted as part of the multimedia block when starting to send new video or audio data.

The video and audio end flag indicates the end of an audio or video transmission. For video

data, parameters and end flag include starting to send data on a new channel or closing a channel

at the end of a video stream. In an exemplary embodiment, audio data has a higher priority than

video data. As a result, multimedia blocks contain all available audio data.

In a preferred embodiment, the sequencer 306 receives the multimedia blocks and

separates them into audio media blocks and video media blocks. The sequencer 306 also uses

the sequence numbers for the multimedia blocks received over the network 100 in order to

ensure the proper ordering of multimedia blocks. The sequencer 306 may temporarily store out

of sequence multimedia blocks pending the receipt of the next anticipated multimedia block. If

20   the missing multimedia block is not received before storage space is exhausted, then the

sequencer 306 assumes the multimedia block is lost.

The audio media blocks are transferred by the room server 110 to the audio resequencer

308 of the transmitter 212. Like the sequencer 306, the audio resequencer 308 puts the audio

data from the audio media blocks into the proper order, i.e., the order in which they were generated. In a preferred embodiment, the audio resequencer 308 differs from the sequencer 306 in that it does not handle packet loss. As a result, it provides more temporary storage for packets that are received out of sequence. From the audio resequencer 308, the sequenced audio media

5   blocks are sent to the multimedia audio queue 312. The multimedia audio queue 312 buffers the audio media blocks until there is available bandwidth at the receiving client 102 to accept additional multimedia data. The audio media blocks are then combined with the video media blocks to form multimedia blocks, which are then sent to the receiving client 102 via the network 100 or any established transmission connection.

10   The room server 110 transfers video media blocks to a video resequencer 310. In a preferred embodiment, there is one video resequencer for each of eight video channels. Each channel handles video data displayed in a unique display window on the display 404 of the client 102. Thus, in the exemplary embodiment with eight video channels, there may be up to eight simultaneously displayed video streams. The video media blocks are transferred to the video

15   multiplexer 314.

The video multiplexer 314 contains a video queue for each video channel. The video queues are FIFO (first in first out) and store video fragments. The video fragment may be a whole video frame, a start of a video frame, a middle of a video frame, an end of a video frame, or a special value that represents a lost video fragment. In a preferred embodiment, only certain

20   sequences of video fragments may be input into the video queue. For example, a 'start' may be followed by a 'middle,' which may be followed by an 'end,' however, a 'start' may not be followed by another 'start.' An entire video frame or a certain number of bytes of a video frame may be output from the video queue. As an example, if a video queue were storing a 200-byte

'start' fragment, then the queue may output, on request, a 100-byte 'start' fragment, leaving a

100-byte 'middle' fragment as the next fragment in the queue.

The video queue in the video multiplexer 314 functions as a buffer for the video data. As

video media blocks are received in order by the video multiplexer 314, they are assembled into

5   complete video frames in the video queue. Once an entire video frame has been assembled, if

there is no available bandwidth in the connection to the receiving client 102 for accepting the

video data, the video queue drops the frame. As bandwidth becomes available in the connection

to the receiving client 102, video media blocks are sent to packet encoder 316 where they are

combined with the audio media blocks to form multimedia blocks. The multimedia blocks are

10   sent to the receiving client 102 via network 100 or via any established transmission connection.

Figure 4 is a block diagram of a client 102 according to an exemplary embodiment of the

present invention. In one embodiment, the client 102 includes a receiver 202, a transmitter 204,

a display 404, a speaker 406, a camera 408, and a microphone 410. Each client 102 is capable of

both transmitting and receiving multimedia data.

15   On the transmitting side, the camera 408 generates video events and the microphone 410

generates audio events. The video events are sent to the video multiplexer 314. Like the video

multiplexer 314 at the room server 110, the video multiplexer 314 at the client has multiple

channels to handle multiple video signals. Thus, the client 102 may contain multiple video

cameras. Also like the video mulitplexer 314 at the room server 110, the video multiplexer 314

20   at the client 102 contains a video queue for each channel, which is used for sequencing and

dropping video frames to reduce bandwidth requirement.

The audio events are sent from the microphone 410 to the multimedia audio queue 312.

As bandwidth becomes available to send the data, video media blocks and audio media blocks

are sent to packet encoder 316 where they are combined to form multimedia blocks. The multimedia blocks are sent to the room server 110 via the network 100, or any established transmission connection.

On the receiving side, the receiver 202 receives multimedia blocks via the network 100 from the room server 110. The sequencer 306 in the receiver 202 orders the multimedia blocks into the proper order and separates them into video media blocks and audio media blocks. The audio media blocks are sent to the speakers 406 where they are converted to into sound, which may be generated in either analog or digital form depending on the particular implementation. The video media blocks are sent to the video demultiplexer 402 where they are broken down into individual video frames. Similar to video multiplexer 314, video demultiplexer 402 contains a video queue that is used for assembling video frames and dropping video frames. The video frames are sent to the video display 404 where they are displayed in a conventional manner.

Figure 5 is a diagram of a threading model according to an exemplary embodiment of the present invention. In addition to multimedia transmissions, receivers 210 and transmitters 212 in the room server 110 also send and receive requests to and from their respective clients 102. These events may include requests to send audio or video to specific clients, request to view the video of specific clients, requests to block clients from viewing video, etc. Clients that are assigned the position of moderator may make requests that are limited to the moderator. Examples of these requests include requests to eject a client, requests to set the privileges of certain clients to have access to certain data types, requests to close a room, or requests to make another client assume the position of moderator.

In a preferred embodiment as shown in Figure 5, a request processor 500 includes an input event thread pool 502, a main thread pool 504, an output event thread pool 506 and a

request queue 508. The input event thread pool 502 is connected to the receiver 210 and the

request queue 508. The request queue 508 is connected to the input event thread pool 502, the

main thread pool 504, and the output event thread pool. The main thread pool 504 is connected

to the request queue 508. The output event thread pool 506 is connected to the request queue

5   508 and the transmitter 212. The request processor 500 may be software code stored in a

memory and executed by a computer processor, although the invention is not limited to this

embodiment. In a preferred embodiment, the memory and computer processor are components

of the room server 110. The software instructions may be stored on a computer-readable

medium, such as a floppy disk, CD ROM, or any other appropriate storage medium. The

10   connections of the components in the request processor 500 may be logical connections defined

by the software code.

The receiver 210 sends input requests received from client 102 to the request processor

500. The input requests are sent to the input event thread pool 502 for processing. Input

requests include request that require an immediate response and long term actions. Input

15   requests that require an immediate response are created to handle incoming network traffic sent

via TCP. Input requests that are long term actions are created to handle incoming network traffic

sent via UDP, if the connection supports UDP as a transmission protocol.

Output requests are sent to the output event thread pool 506 for processing. Output

requests are created to handle outbound data sent via UDP, if the connection supports UDP as a

20   transmission protocol. In processing the output request, the output event thread pool 506

generates an output event. This event calls one or more transmitters 212 to send outbound data

to clients 102.

Internal requests are used to perform tasks that are internal to the room server 110. Internal requests consist of retransmission of audio and video within the room server 110, as well as other tasks which are not appropriate to handle in an input or output request because of potential locking or blocking issues. Internal requests are stored in request queue 508, and are

5   dispatched to the main thread pool 504 as threads become available.

Figure 6 is a flow chart of dynamic data transmission according to an exemplary embodiment of the present invention. The process of dynamic data transmission is facilitated by both the client 102 and room server 110 to ensure minimum latency in the transmission and receipt of multimedia data. When a client 102 initiates a conferencing session by logging-in

10   through an entry server 106, a bandwidth regulator determines 602 the current bandwidth and latency for outgoing and incoming multimedia transmissions. The clients 102 and room servers 110 each contain bandwidth regulators, which, in a preferred embodiment, are implemented in software. Based on the bandwidth and latency information, the bandwidth regulator determines 604 the optimal packet size and optimal packet interval for each connection. The room server

15   110 records 606, in a journal, the packet size and departure time for the next packet sent by transmitter 212. The client 102 sends 606 the next packet and records, in a journal, the packet size and departure time for this packet.

In one embodiment, the sender (either the room server 110 or the client 102) then determines 608 whether there is more data to be sent to the receiver. If there is no more data to

20   be sent, the process ends. If there is additional data to be sent, then the bandwidth regulator updates 610 the journal by removing records from that journal for each receipt received from the inbound multimedia stream. At the room server 110, the receipts will be accepted at receiver 210. At client 102, the receipts will be accepted at receiver 202. The bandwidth regulator also

removes records from the journal for packets that have been lost. The bandwidth regulator then determines 612 the expected arrival time for the receipts corresponding to each remaining entry in the journal. The expected arrival time is determined by using the departure time of the packet, the latency, and the outbound and inbound packet size and bandwidth.

5   The bandwidth regulators at client 102 and room server 110 then uses the expected arrival time to determine 614 whether any journaled packets are overdue. If there are overdue packets, then the bandwidth regulator enters 616 a mode in which transmitter 204, 212 sends only audio data. Since the audio data requires lower bandwidth for transmission than video and audio data combined, the latency of the transmission will decrease if the data is limited to only

10   audio. If there are no overdue packets, then the bandwidth regulator enters 618 a mode in which transmitter 204, 212 sends both audio and video data. If there is enough available bandwidth in the connection to handle video and audio data, there will be no overdue packets and the bandwidth regulator will allow the transmission of both audio and video data. The result of switching between these two modes is that, for lower bandwidth connections, audio data is sent

15   continuously with intermittent transmissions of video data. Once either the audio mode or audio and video mode has been entered, the client 102 or room server 110 sends 606 the next packet and records the packet size and departure time for this packet.

          Having fully described a preferred embodiment of the invention and various alternatives,

20   those skilled in the art will recognize, given the teachings herein, that numerous alternatives and equivalents exist which do not depart from the invention. It is therefore intended that the invention not be limited by the foregoing description, but only by the appended claims.